**Typing Speed Checker**

**TSDL-Python**

**Guided By: Vaishali Mam**

GROUP MEMBERS NAME AND ROLL NO

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**CONTENTS**

1. Introduction ……………………… page no.1

2. Methodology ……………………… page no.1

3. Flowchart ……………………… Page no. 2

4. Implementation……………………… Page no.3

5. Results ………………………. Page no. 6

6 . Conclusion…………………………..Page no. 7

**INTRODUCTION :**

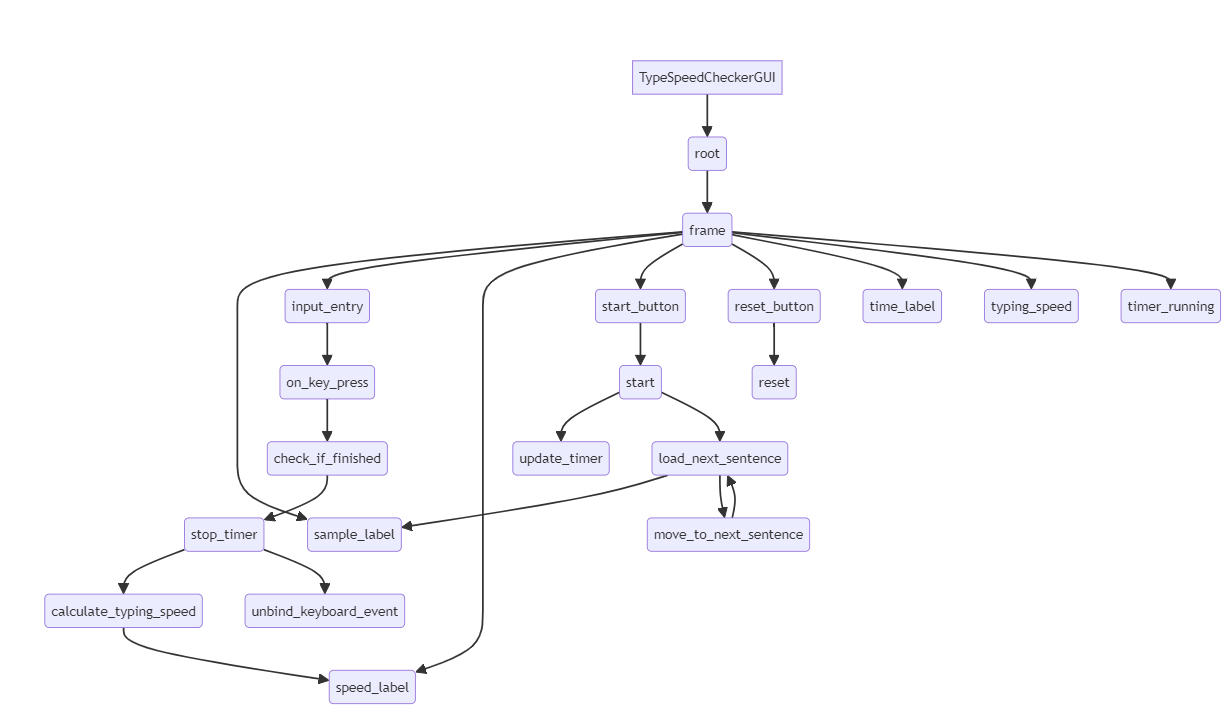
A typing speed checker is a software application that measures the speed and accuracy of a user's typing. It is typically used to improve typing skills and to assess typing proficiency for employment purposes. Typing speed checkers can be implemented in various programming languages, including Python**.**

**METHODOLOGY :**

A Python typing speed checker typically works by generating a random passage of text for the user to type. The time it takes the user to type the passage is recorded, and the typing speed is calculated in words per minute (WPM). To calculate accuracy, the typing speed checker compares the user's typed text to the original passage and counts the number of errors.

* Add gamification elements. This can make the typing speed checker more engaging and motivating for the user. For example, you could add a leaderboard, badges, or points system.
* Support multiple languages. This would allow users from all over the world to use the typing speed checker.
* Integrate with other applications. For example, you could integrate the typing speed checker with a word processor or online learning platform.

**FLOWCHART :**

****

**IMPLEMENTATION :**

import tkinter as tk

import time

class TypeSpeedCheckerGUI:

    def \_\_init\_\_(self):

        self.root = tk.Tk()

        self.root.title("Typing Speed Checker")

        self.root.geometry("800x400")

        self.frame = tk.Frame(self.root)

        self.current\_time = time.time()

        # List of sentences for the user to type

        self.sentences = [

            "Type this text as fast as you can.",

            "The quick brown fox jumps over the lazy dog.",

            "To be or not to be, that is the question.",

            "A journey of a thousand miles begins with a single step.",

            "All that glitters is not gold."

        ]

        self.current\_sentence\_index = 0

        # Create a label to display the text to type

        self.sample\_label = tk.Label(self.frame, text="", font=("Helvetica", 18))

        self.sample\_label.grid(row=0, column=0, padx=5, pady=10)

        # Create an entry for the user to type the text

        self.input\_entry = tk.Entry(self.frame, width=40, font=("Helvetica", 24))

        self.input\_entry.grid(row=1, column=0, padx=5, pady=10)

        # Create a label to display the user's typing speed

        self.speed\_label = tk.Label(self.frame, text="Speed:\n0.00 CPS\n0.00 CPM", font=("Helvetica", 18))

        self.speed\_label.grid(row=2, column=0, padx=5, pady=10)

        # Create a button to start the typing test

        self.start\_button = tk.Button(self.frame, text="Start", command=self.start)

        self.start\_button.grid(row=3, column=0, padx=5, pady=10)

        # Create a button to reset the typing test

        self.reset\_button = tk.Button(self.frame, text="Reset", command=self.reset)

        self.reset\_button.grid(row=3, column=1, padx=5, pady=10)

        self.frame.pack(expand=True)

        self.time\_label = tk.Label(self.frame, text="Time elapsed:\n0.00 seconds", font=("Helvetica", 18))

        self.time\_label.grid(row=2, column=1, padx=5, pady=10)

        # Initialize the typing speed

        self.typing\_speed = 0.0

        self.timer\_running = False

        # Start the mainloop

        self.root.mainloop()

    def start(self):

        if not self.timer\_running:

            # Start the timer

            self.start\_time = time.time()

            self.timer\_running = True

            self.update\_timer()

            self.load\_next\_sentence()  # Load the first sentence

        # Bind the keyboard event to the input entry

        self.input\_entry.bind("<KeyPress>", self.on\_key\_press)

    def reset(self):

        if self.timer\_running:

            self.timer\_running = False

            # Reset the timer and the time label

            self.start\_time = time.time()

            self.current\_time = time.time()

            self.time\_label.config(text="Time elapsed:\n0.00 seconds")

            # Clear the input entry

            self.input\_entry.delete(0, "end")

            # Reset the typing speed

            self.typing\_speed = 0.0

            # Reset the speed label

            self.speed\_label.config(text="Speed:\n0.00 CPS\n0.00 CPM")

            # Unbind the keyboard event

            self.input\_entry.unbind("<KeyPress")

    def update\_timer(self):

        if self.timer\_running:

            elapsed\_time = time.time() - self.start\_time

            self.time\_label.config(text=f"Time elapsed:\n{elapsed\_time:.2f} seconds")

            self.root.after(100, self.update\_timer)

    def load\_next\_sentence(self):

        if self.current\_sentence\_index < len(self.sentences):

            self.sample\_label.config(text=self.sentences[self.current\_sentence\_index])

        else:

            # All sentences have been completed

            self.sample\_label.config(text="All sentences completed")

    def on\_key\_press(self, event):

        # Check if the user has finished typing the text

        if self.input\_entry.get() == self.sentences[self.current\_sentence\_index]:

            self.timer\_running = False

            # Calculate the typing speed

            self.typing\_speed = len(self.input\_entry.get()) / (time.time() - self.start\_time)

            # Display the typing speed

            self.speed\_label.config(text=f"Speed:\n{self.typing\_speed:.2f} CPS\n{self.typing\_speed \* 60:.2f} CPM")

            # Unbind the keyboard event

            self.input\_entry.unbind("<KeyPress")

            # Move to the next sentence

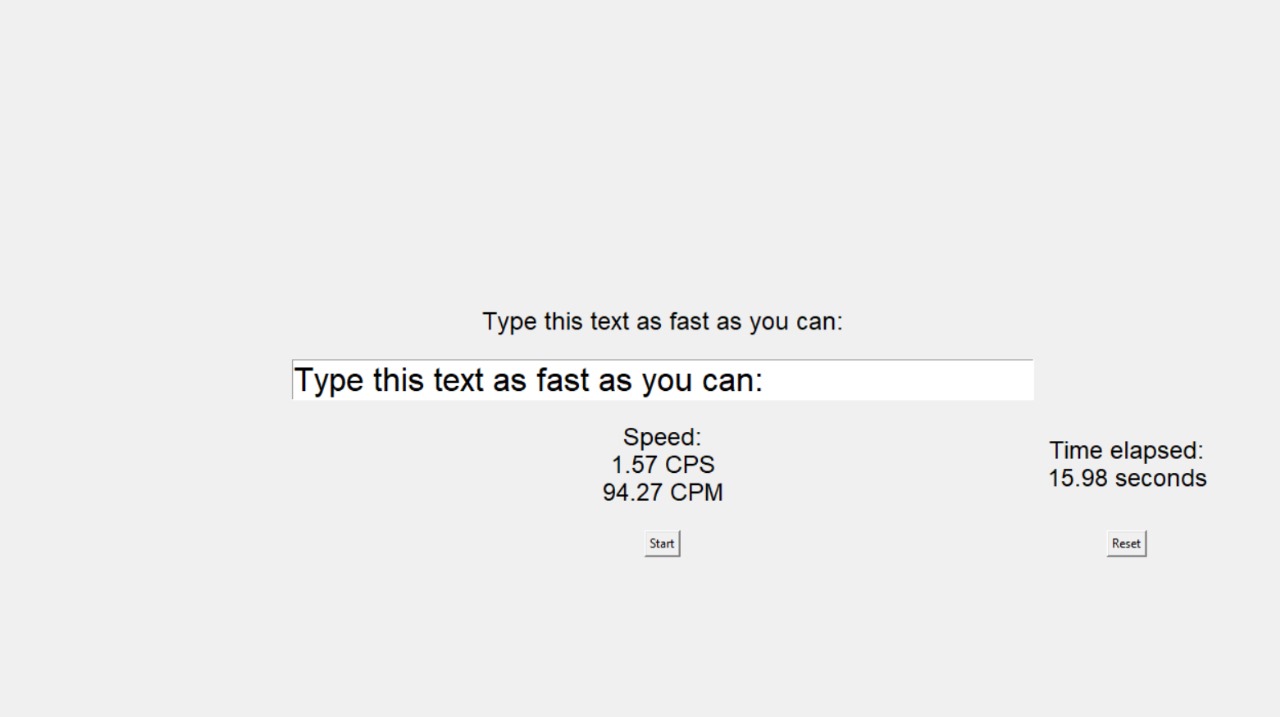
            self.current\_sentence\_index += 1

            self.load\_next\_sentence()

if \_\_name\_\_ == "\_\_main\_\_":

    TypeSpeedCheckerGUI()

**RESULTS :**



**CONCLUSION**

Python typing speed checkers are a simple and effective way to improve typing skills and to assess typing proficiency. They are easy to implement and can be customized to meet the specific needs of the user.